

Welcome to E-XFL.COM

#### What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

E·XFI

Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	40 MIPs
Connectivity	I <sup>2</sup> C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	85
Program Memory Size	128KB (43K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 32x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24hj128gp310-i-pt

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

#### Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION. QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

## QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV ISO/TS 16949:2002

#### Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, rfPIC, SmartShunt and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, nanoWatt XLP, PICkit, PICDEM, PICDEM.net, PICtail, PIC<sup>32</sup> logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Total Endurance, TSHARC, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

 $\ensuremath{\mathsf{SQTP}}$  is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2009, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.



Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and mulfacture of development systems is ISO 9001:2000 certified.

### Pin Diagrams (Continued)





Vector Number	Interrupt Request (IRQ) Number	IVT Address	AIVT Address	Interrupt Source
8	0	0x000014	0x000114	INT0 – External Interrupt 0
9	1	0x000016	0x000116	IC1 – Input Compare 1
10	2	0x000018	0x000118	OC1 – Output Compare 1
11	3	0x00001A	0x00011A	T1 – Timer1
12	4	0x00001C	0x00011C	DMA0 – DMA Channel 0
13	5	0x00001E	0x00011E	IC2 – Input Capture 2
14	6	0x000020	0x000120	OC2 – Output Compare 2
15	7	0x000022	0x000122	T2 – Timer2
16	8	0x000024	0x000124	T3 – Timer3
17	9	0x000026	0x000126	SPI1E – SPI1 Error
18	10	0x000028	0x000128	SPI1 – SPI1 Transfer Done
19	11	0x00002A	0x00012A	U1RX – UART1 Receiver
20	12	0x00002C	0x00012C	U1TX – UART1 Transmitter
21	13	0x00002E	0x00012E	ADC1 – Analog-to-Digital Converter 1
22	14	0x000030	0x000130	DMA1 – DMA Channel 1
23	15	0x000032	0x000132	Reserved
24	16	0x000034	0x000134	SI2C1 – I2C1 Slave Events
25	17	0x000036	0x000136	MI2C1 – I2C1 Master Events
26	18	0x000038	0x000138	Reserved
27	19	0x00003A	0x00013A	CN - Change Notification Interrupt
28	20	0x00003C	0x00013C	INT1 – External Interrupt 1
29	21	0x00003E	0x00013E	ADC2 – Analog-to-Digital Converter 2
30	22	0x000040	0x000140	IC7 – Input Capture 7
31	23	0x000042	0x000142	IC8 – Input Capture 8
32	24	0x000044	0x000144	DMA2 – DMA Channel 2
33	25	0x000046	0x000146	OC3 – Output Compare 3
34	26	0x000048	0x000148	OC4 – Output Compare 4
35	27	0x00004A	0x00014A	T4 – Timer4
36	28	0x00004C	0x00014C	T5 – Timer5
37	29	0x00004E	0x00014E	INT2 – External Interrupt 2
38	30	0x000050	0x000150	U2RX – UART2 Receiver
39	31	0x000052	0x000152	U2TX – UART2 Transmitter
40	32	0x000054	0x000154	SPI2E – SPI2 Error
41	33	0x000056	0x000156	SPI1 – SPI1 Transfer Done
42	34	0x000058	0x000158	C1RX – ECAN1 Receive Data Ready
43	35	0x00005A	0x00015A	C1 – ECAN1 Event
44	36	0x00005C	0x00015C	DMA3 – DMA Channel 3
45	37	0x00005E	0x00015E	IC3 – Input Capture 3
46	38	0x000060	0x000160	IC4 – Input Capture 4
47	39	0x000062	0x000162	IC5 – Input Capture 5
48	40	0x000064	0x000164	IC6 – Input Capture 6
49	41	0x000066	0x000166	OC5 – Output Compare 5
50	42	0x000068	0x000168	OC6 – Output Compare 6
51	43	0x00006A	0x00016A	OC7 – Output Compare 7
52	44	0x00006C	0x00016C	OC8 – Output Compare 8
53	45	0x00006E	0x00016E	Reserved

TABLE 7-1:INTERRUPT VECTORS

### REGISTER 7-17: IPC2: INTERRUPT PRIORITY CONTROL REGISTER 2

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
_		U1RXIP<2:0>				SPI1IP<2:0>	
bit 15							bit 8
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
		SPI1EIP<2:0>				T3IP<2:0>	
bit 7							bit 0
Logond							
R = Readable	hit	W = Writable h	nit	II = Unimpler	mented hit rea	ad as '0'	
-n = Value at P	OR	'1' = Rit is set		0' = Bit is cle	ared	x = Bit is unkn	own
II Value at I		1 Dit lo oot					
bit 15	Unimpleme	ented: Read as 'o	,				
bit 14-12	U1RXIP<2:	0>: UART1 Recei	iver Interrupt	Priority bits			
	111 = Interr	rupt is priority 7 (h	ighest priorit	y interrupt)			
	•						
	•						
	001 = Interr	upt is priority 1					
	000 = Interr	upt source is disa	abled				
bit 11	Unimpleme	ented: Read as 'o	,				
bit 10-8	SPI1IP<2:0	>: SPI1 Event Int	errupt Priority	/ bits			
	111 = Interr	rupt is priority 7 (h	ighest priorit	y interrupt)			
	•						
	•						
	001 = Interr 000 = Interr	upt is priority 1 upt source is disa	abled				
bit 7	Unimpleme	ented: Read as 'o	,				
bit 6-4	SPI1EIP<2:	0>: SPI1 Error In	terrupt Priorit	y bits			
	111 = Interr	rupt is priority 7 (h	ighest priorit	y interrupt)			
	•						
	•						
	001 = Interr 000 = Interr	upt is priority 1 upt source is disa	abled				
bit 3	Unimpleme	ented: Read as 'o	,				
bit 2-0	T3IP<2:0>:	Timer3 Interrupt	Priority bits				
	111 = Interr	upt is priority 7 (h	ighest priorit	y interrupt)			
	•						
	•						
	001 = Interr	upt is priority 1					
	000 = Interr	upt source is disa	abled				

### REGISTER 7-25: IPC10: INTERRUPT PRIORITY CONTROL REGISTER 10

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—		OC7IP<2:0>		—		OC6IP<2:0>	
bit 15					•		bit 8
U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
		OC5IP<2:0>		_		IC6IP<2:0>	
bit 7							bit 0
Lonordi							
Legena:	bit	W - Writabla k	sit	II – Unimploi	montod bit rog	ad as '0'	
R = Reducible bit $W = Wi$		'1' - Bit is sot	JIL	$0^{\circ} - 0^{\circ}$	nenieu bil, rea	v – Bitic unkn	014/D
		I – Dit is set					OWIT
bit 15	Unimpleme	ented: Read as 'o	,				
bit 14-12	OC7IP<2:0	>: Output Compa	re Channel 7	Interrupt Prior	itv bits		
	111 = Interr	upt is priority 7 (h	nighest priorit	y interrupt)			
	•						
	•						
	001 = Interr	upt is priority 1					
	000 = Interr	upt source is disa	abled				
bit 11	Unimpleme	ented: Read as 'c	)'				
bit 10-8	OC6IP<2:0	: Output Compa	re Channel 6	Interrupt Prior	ity bits		
	111 = Interr	rupt is priority 7 (h	nighest priorit	y interrupt)			
	•						
	•						
	001 = Interr 000 = Interr	upt is priority 1 upt source is disa	abled				
bit 7	Unimpleme	ented: Read as 'c	)'				
bit 6-4	OC5IP<2:0	>: Output Compa	re Channel 5	Interrupt Prior	ity bits		
	111 = Interr	upt is priority 7 (h	nighest priorit	y interrupt)			
	•						
	•						
	001 = Interr 000 = Interr	upt is priority 1 upt source is disa	abled				
bit 3	Unimpleme	nted: Read as 'c	)'				
bit 2-0	IC6IP<2:0>	: Input Capture C	hannel 6 Inte	errupt Priority b	its		
	111 = Interr	upt is priority 7 (h	nighest priorit	y interrupt)			
	•						
	•						
	001 = Interr	upt is priority 1					
	000 = Interr	upt source is disa	abled				

**CLKDIV: CLOCK DIVISOR REGISTER** 

**REGISTER 9-2:** 

#### R/W-0 R/W-0 R/W-1 **R/W-1** R/W-0 R/W-0 R/W-0 R/W-0 DOZEN<sup>(1)</sup> ROI DOZE<2:0> FRCDIV<2:0> bit 15 bit 8 R/W-0 R/W-1 U-0 R/W-0 R/W-0 R/W-0 R/W-0 R/W-0 PLLPRE<4:0> PLLPOST<1:0> bit 7 bit 0 Legend: y = Value set from Configuration bits on POR R = Readable bit W = Writable bit U = Unimplemented bit, read as '0' -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown bit 15 ROI: Recover on Interrupt bit 1 = Interrupts will clear the DOZEN bit and the processor clock/peripheral clock ratio is set to 1:1 0 = Interrupts have no effect on the DOZEN bit bit 14-12 DOZE<2:0>: Processor Clock Reduction Select bits 000 = FCY/1001 = FCY/2010 = FCY/4011 = FCY/8 (default) 100 = FCY/16101 = FCY/32 110 = FCY/64 111 = FCY/128 DOZEN: DOZE Mode Enable bit<sup>(1)</sup> bit 11 1 = DOZE<2:0> field specifies the ratio between the peripheral clocks and the processor clocks 0 = Processor clock/peripheral clock ratio forced to 1:1 bit 10-8 FRCDIV<2:0>: Internal Fast RC Oscillator Postscaler bits 000 = FRC divide by 1 (default) 001 = FRC divide by 2 010 = FRC divide by 4011 = FRC divide by 8 100 = FRC divide by 16 101 = FRC divide by 32 110 = FRC divide by 64 111 = FRC divide by 256 bit 7-6 PLLPOST<1:0>: PLL VCO Output Divider Select bits (also denoted as 'N2', PLL postscaler) 00 = Output/201 = Output/4 (default) 10 = Reserved 11 = Output/8bit 5 Unimplemented: Read as '0' bit 4-0 PLLPRE<4:0>: PLL Phase Detector Input Divider bits (also denoted as 'N1', PLL prescaler) 00000 = Input/2 (default) 00001 = Input/3 11111 = Input/33

Note 1: This bit is cleared when the ROI bit is set and an interrupt occurs.

### REGISTER 17-1: I2CxCON: I2Cx CONTROL REGISTER (CONTINUED)

bit 5	<b>ACKDT:</b> Acknowledge Data bit (when operating as I <sup>2</sup> C master, applicable during master receive)
	Value that will be transmitted when the software initiates an Acknowledge sequence. 1 = Send NACK during Acknowledge 0 = Send ACK during Acknowledge
bit 4	<b>ACKEN:</b> Acknowledge Sequence Enable bit (when operating as I <sup>2</sup> C master, applicable during master receive)
	<ul> <li>1 = Initiate Acknowledge sequence on SDAx and SCLx pins and transmit ACKDT data bit. Hardware clear at end of master Acknowledge sequence.</li> <li>0 = Acknowledge sequence not in progress</li> </ul>
bit 3	<b>RCEN:</b> Receive Enable bit (when operating as $I^2C$ master)
	1 = Enables Receive mode for $I^2C$ . Hardware clear at end of eighth bit of master receive data byte. 0 = Receive sequence not in progress
bit 2	<b>PEN:</b> Stop Condition Enable bit (when operating as I <sup>2</sup> C master)
	<ul> <li>1 = Initiate Stop condition on SDAx and SCLx pins. Hardware clear at end of master Stop sequence.</li> <li>0 = Stop condition not in progress</li> </ul>
bit 1	<b>RSEN:</b> Repeated Start Condition Enable bit (when operating as I <sup>2</sup> C master)
	<ul> <li>1 = Initiate Repeated Start condition on SDAx and SCLx pins. Hardware clear at end of master Repeated Start sequence.</li> </ul>
	0 = Repeated Start condition not in progress
bit 0	SEN: Start Condition Enable bit (when operating as I <sup>2</sup> C master)
	<ul> <li>1 = Initiate Start condition on SDAx and SCLx pins. Hardware clear at end of master Start sequence.</li> <li>0 = Start condition not in progress</li> </ul>

### 19.0 ENHANCED CAN (ECAN™) MODULE

Note: This data sheet summarizes the features of the PIC24HJXXXGPX06/X08/X10 family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the *"PIC24H Family Reference Manual"*, Section 21. "Enhanced Controller Area Network (ECAN™)" (DS70226), which is available from the Microchip website (www.microchip.com).

### 19.1 Overview

The Enhanced Controller Area Network (ECAN<sup>™</sup>) module is a serial interface, useful for communicating with other CAN modules or microcontroller devices. This interface/protocol was designed to allow communications within noisy environments. The PIC24HJXXXGPX06/X08/X10 devices contain up to two ECAN modules.

The CAN module is a communication controller implementing the CAN 2.0 A/B protocol, as defined in the BOSCH specification. The module will support CAN 1.2, CAN 2.0A, CAN 2.0B Passive and CAN 2.0B Active versions of the protocol. The module implementation is a full CAN system. The CAN specification is not covered within this data sheet. The reader may refer to the BOSCH CAN specification for further details.

The module features are as follows:

- Implementation of the CAN protocol, CAN 1.2, CAN 2.0A and CAN 2.0B
- Standard and extended data frames
- · 0-8 bytes data length
- Programmable bit rate up to 1 Mbit/sec
- Automatic response to remote transmission requests
- Up to 8 transmit buffers with application specified prioritization and abort capability (each buffer may contain up to 8 bytes of data)
- Up to 32 receive buffers (each buffer may contain up to 8 bytes of data)
- Up to 16 full (standard/extended identifier) acceptance filters
- 3 full acceptance filter masks
- DeviceNet<sup>™</sup> addressing support
- Programmable wake-up functionality with integrated low-pass filter
- Programmable Loopback mode supports self-test operation
- Signaling via interrupt capabilities for all CAN receiver and transmitter error states
- · Programmable clock source
- Programmable link to input capture module (IC2 for both CAN1 and CAN2) for time-stamping and

network synchronization

· Low-power Sleep and Idle mode

The CAN bus module consists of a protocol engine and message buffering/control. The CAN protocol engine handles all functions for receiving and transmitting messages on the CAN bus. Messages are transmitted by first loading the appropriate data registers. Status and errors can be checked by reading the appropriate registers. Any message detected on the CAN bus is checked for errors and then matched against filters to see if it should be received and stored in one of the receive registers.

### 19.2 Frame Types

The CAN module transmits various types of frames which include data messages, remote transmission requests and as other frames that are automatically generated for control purposes. The following frame types are supported:

· Standard Data Frame:

A standard data frame is generated by a node when the node wishes to transmit data. It includes an 11-bit standard identifier (SID) but not an 18-bit extended identifier (EID).

Extended Data Frame:

An extended data frame is similar to a standard data frame but includes an extended identifier as well.

· Remote Frame:

It is possible for a destination node to request the data from the source. For this purpose, the destination node sends a remote frame with an identifier that matches the identifier of the required data frame. The appropriate data source node will then send a data frame as a response to this remote request.

Error Frame:

An error frame is generated by any node that detects a bus error. An error frame consists of two fields: an error flag field and an error delimiter field.

· Overload Frame:

An overload frame can be generated by a node as a result of two conditions. First, the node detects a dominant bit during interframe space which is an illegal condition. Second, due to internal conditions, the node is not yet able to start reception of the next message. A node may generate a maximum of 2 sequential overload frames to delay the start of the next message.

· Interframe Space:

Interframe space separates a proceeding frame (of whatever type) from a following data or remote frame.

REGISTER 19-16:	CIRXFnSID: ECAN™ MODULE ACCEPTANCE FILTER n STANDARD IDENTIFIER
	(n = 0, 1,, 15)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
bit 15							bit 8

R/W-x	R/W-x	R/W-x	U-0	R/W-x	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDE	—	EID17	EID16
bit 7							bit 0

Legend:				
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'		
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown	

bit 15-5	SID<10:0>: Standard Identifier bits 1 = Message address bit SIDx must be '1' to match filter 0 = Message address bit SIDx must be '6' to match filter
<b>h</b> :+ 4	U = Message address bit SIDX must be 0 to match litter
DIC 4	Unimplemented: Read as 0
bit 3	EXIDE: Extended Identifier Enable bit
	If MIDE = 1 then:
	1 = Match only messages with extended identifier addresses
	0 = Match only messages with standard identifier addresses
	If MIDE = 0 then:
	Ignore EXIDE bit.
bit 2	Unimplemented: Read as '0'
bit 1-0	EID<17:16>: Extended Identifier bits
	1 = Message address bit EIDx must be '1' to match filter
	0 = Message address bit EIDx must be '0' to match filter

# REGISTER 19-17: CIRXFnEID: ECAN™ MODULE ACCEPTANCE FILTER n EXTENDED IDENTIFIER (n = 0, 1, ..., 15)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 15							bit 8

| R/W-x |
|-------|-------|-------|-------|-------|-------|-------|-------|
| EID7  | EID6  | EID5  | EID4  | EID3  | EID2  | EID1  | EID0  |
| bit 7 |       |       |       |       |       |       | bit 0 |

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read	1 as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 EID<15:0>: Extended Identifier bits

1 = Message address bit EIDx must be '1' to match filter

0 = Message address bit EIDx must be '0' to match filter

© 2009 Microchip Technology Inc.

### REGISTER 19-24: CIRXOVF1: ECAN™ MODULE RECEIVE BUFFER OVERFLOW REGISTER 1

R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
RXOVF15	RXOVF14	RXOVF13	RXOVF12	RXOVF11	RXOVF10	RXOVF9	RXOVF8
bit 15							bit 8

| R/C-0  |
|--------|--------|--------|--------|--------|--------|--------|--------|
| RXOVF7 | RXOVF6 | RXOVF5 | RXOVF4 | RXOVF3 | RXOVF2 | RXOVF1 | RXOVF0 |
| bit 7  |        |        |        |        |        |        | bit 0  |

Legend:	C = Clear only bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read	d as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **RXOVF<15:0>:** Receive Buffer n Overflow bits

1 = Module pointed a write to a full buffer (set by module)

0 = Overflow is cleared (clear by application software)

### REGISTER 19-25: CIRXOVF2: ECAN™ MODULE RECEIVE BUFFER OVERFLOW REGISTER 2

| R/C-0   |
|---------|---------|---------|---------|---------|---------|---------|---------|
| RXOVF31 | RXOVF30 | RXOVF29 | RXOVF28 | RXOVF27 | RXOVF26 | RXOVF25 | RXOVF24 |
| bit 15  |         |         |         |         |         |         | bit 8   |

| R/C-0   |
|---------|---------|---------|---------|---------|---------|---------|---------|
| RXOVF23 | RXOVF22 | RXOVF21 | RXOVF20 | RXOVF19 | RXOVF18 | RXOVF17 | RXOVF16 |
| bit 7   |         |         |         |         |         |         | bit 0   |

Legend:	C = Clear only bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read	d as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **RXOVF<31:16>:** Receive Buffer n Overflow bits

1 = Module pointed a write to a full buffer (set by module)

0 = Overflow is cleared (clear by application software)

#### REGISTER 19-29: CiTRBnDLC: ECAN™ MODULE BUFFER n DATA LENGTH CONTROL (n = 0, 1, ..., 31)

	( )						
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID5	EID4	EID3	EID2	EID1	EID0	RTR	RB1
bit 15							bit 8

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
_	—	—	RB0	DLC3	DLC2	DLC1	DLC0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read	as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-10	EID<5:0>: Extended Identifier bits
bit 9	RTR: Remote Transmission Request bit
	<ul><li>1 = Message will request remote transmission</li><li>0 = Normal message</li></ul>
bit 8	RB1: Reserved Bit 1
	User must set this bit to '0' per CAN protocol.
bit 7-5	Unimplemented: Read as '0'
bit 4	RB0: Reserved Bit 0
	User must set this bit to '0' per CAN protocol.
bit 3-0	DLC<3:0>: Data Length Code bits

# REGISTER 19-30: CiTRBnDm: ECAN<sup>TM</sup> MODULE BUFFER n DATA FIELD BYTE m $(n = 0, 1, ..., 31; m = 0, 1, ..., 7)^{(1)}$

| R/W-x   |
|---------|---------|---------|---------|---------|---------|---------|---------|
| TRBnDm7 | TRBnDm6 | TRBnDm5 | TRBnDm4 | TRBnDm3 | TRBnDm2 | TRBnDm1 | TRBnDm0 |
| bit 7   |         |         |         |         |         |         | bit 0   |

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read	d as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

### bit 7-0 TRnDm<7:0>: Data Field Buffer 'n' Byte 'm' bits

Note 1: The Most Significant Byte contains byte (m + 1) of the buffer.

'1' = Bit is set

### REGISTER 19-31: CITRBnSTAT: ECAN™ MODULE RECEIVE BUFFER n STATUS

	(n = 0,	1,, 31)							
U-0 U-0		U-0	U-0 R/W-x		R/W-x	R/W-x	R/W-x		
_		_	FILHIT4	FILHIT3	FILHIT2	FILHIT1	FILHIT0		
bit 15							bit 8		
U-0	U-0	U-0 U-0 U-0		U-0	U-0	U-0	U-0		
—	—	—	_			—	—		
bit 7							bit 0		
Legend:									
R = Readable bit W = Writable bit			bit	U = Unimplemented bit, read as '0'					

bit 15-13 **Unimplemented:** Read as '0'

-n = Value at POR

bit 12-8 **FILHIT<4:0>:** Filter Hit Code bits (only written by module for receive buffers, unused for transmit buffers) Encodes number of filter that resulted in writing this buffer.

'0' = Bit is cleared

bit 7-0 Unimplemented: Read as '0'

x = Bit is unknown





Field	Description					
Wnd     One of 16 destination working registers ∈ {W0W15}						
Wns One of 16 source working registers ∈ {W0W15}						
WREG	W0 (working register used in file register instructions)					
Ws         Source W register ∈ { Ws, [Ws], [Ws++], [Ws], [++Ws], [Ws] }						
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns], [++Wns], [Wns], [Wns+Wb] }					

### TABLE 22-1: SYMBOLS USED IN OPCODE DESCRIPTIONS (CONTINUED)

### TABLE 22-2: INSTRUCTION SET OVERVIEW

Base Instr #	Assembly Mnemonic		Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
1	1 ADD		f	f = f + WREG	1	1	C,DC,N,OV,Z
			f,WREG	WREG = f + WREG	1	1	C,DC,N,OV,Z
		ADD	#lit10,Wn	Wd = lit10 + Wd	1	1	C,DC,N,OV,Z
		ADD	Wb,Ws,Wd	Wd = Wb + Ws	1	1	C,DC,N,OV,Z
		ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C,DC,N,OV,Z
2	ADDC	ADDC	f	f = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	f,WREG	WREG = $f + WREG + (C)$	1	1	C,DC,N,OV,Z
		ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C,DC,N,OV,Z
3	AND	AND	f	f = f .AND. WREG	1	1	N,Z
		AND	f,WREG	WREG = f .AND. WREG	1	1	N,Z
		AND	#lit10,Wn	Wd = lit10 .AND. Wd	1	1	N,Z
		AND	Wb,Ws,Wd	Wd = Wb .AND. Ws	1	1	N,Z
		AND	Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N,Z
4	ASR	ASR	f	f = Arithmetic Right Shift f	1	1	C,N,OV,Z
		ASR	f,WREG	WREG = Arithmetic Right Shift f	1	1	C,N,OV,Z
		ASR	Ws,Wd	Wd = Arithmetic Right Shift Ws	1	1	C,N,OV,Z
		ASR	Wb,Wns,Wnd	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N,Z
		ASR	Wb,#lit5,Wnd	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N,Z
5	BCLR	BCLR	f,#bit4	Bit Clear f	1	1	None
		BCLR	Ws,#bit4	Bit Clear Ws	1	1	None
6	BRA	BRA	C,Expr	Branch if Carry	1	1 (2)	None
		BRA	GE,Expr	Branch if greater than or equal	1	1 (2)	None
		BRA	GEU,Expr	Branch if unsigned greater than or equal	1	1 (2)	None
		BRA	GT,Expr	Branch if greater than	1	1 (2)	None
		BRA	GTU, Expr	Branch if unsigned greater than	1	1 (2)	None
		BRA	LE, Expr	Branch if less than or equal	1	1 (2)	None
		BRA	LEU,Expr	Branch if unsigned less than or equal	1	1 (2)	None
		BRA	LT,Expr	Branch if less than	1	1 (2)	None
		BRA	LTU, Expr	Branch if unsigned less than	1	1 (2)	None
		BRA	N,Expr	Branch if Negative	1	1 (2)	None
		BRA	NC,Expr	Branch if Not Carry	1	1 (2)	None
		BRA	NN, Expr	Branch if Not Negative	1	1 (2)	None
		BRA	NZ,Expr	Branch if Not Zero	1	1 (2)	None
		BRA	Expr	Branch Unconditionally	1	2	None
		BRA	Z,Expr	Branch if Zero	1	1 (2)	None
		BRA	Wn	Computed Branch	1	2	None
7	BSET	BSET	f,#bit4	Bit Set f	1	1	None
		BSET	Ws,#bit4	Bit Set Ws	1	1	None
8	BSW	BSW.C	Ws,Wb	Write C bit to Ws <wb></wb>	1	1	None
		BSW.Z	Ws,Wb	Write Z bit to Ws <wb></wb>	1	1	None
9	BTG	BTG	f,#bit4	Bit Toggle f	1	1	None
		BTG	Ws,#bit4	Bit Toggle Ws	1	1	None
10	BTSC	BTSC	f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
		BTSC	Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None
11	BTSS	BTSS	f,#bit4	Bit Test f, Skip if Set	1	1 (2 or 3)	None
		BTSS	Ws,#bit4	Bit Test Ws, Skip if Set	1	1 (2 or 3)	None

## 23.0 DEVELOPMENT SUPPORT

The PIC<sup>®</sup> microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM<sup>™</sup> Assembler
  - MPLAB C18 and MPLAB C30 C Compilers
  - MPLINK<sup>™</sup> Object Linker/
  - MPLIB™ Object Librarian
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
  - PICkit<sup>™</sup> 2 Development Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

### 23.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit microcontroller market. The MPLAB IDE is a Windows<sup>®</sup> operating system-based application that contains:

- · A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- · A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Visual device initializer for easy register initialization
- · Mouse over variable inspection
- Drag and drop variables from source to watch windows
- · Extensive on-line help
- Integration of select third party tools, such as HI-TECH Software C Compilers and IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PIC MCU emulator and simulator tools (automatically updates all project information)
- · Debug using:
  - Source files (assembly or C)
  - Mixed assembly and C
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

## 24.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of PIC24HJXXXGPX06/X08/X10 electrical characteristics. Additional information is provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the PIC24HJXXXGPX06/X08/X10 family are listed below. Exposure to these maximum rating conditions for extended periods can affect device reliability. Functional operation of the device at these or any other conditions above the parameters indicated in the operation listings of this specification is not implied.

### Absolute Maximum Ratings<sup>(1)</sup>

Ambient temperature under bias	40°C to +85°C
Storage temperature	65°C to +150°C
Voltage on VDD with respect to Vss	0.3V to +4.0V
Voltage on any combined analog and digital pin and MCLR, with respect to Vss	0.3V to (VDD + 0.3V)
Voltage on any digital-only pin with respect to Vss	0.3V to +5.6V
Voltage on VCAP/VDDCORE with respect to Vss	2.25V to 2.75V
Maximum current out of Vss pin	
Maximum current into VDD pin <sup>(2)</sup>	250 mA
Maximum output current sunk by any I/O pin <sup>(3)</sup>	4 mA
Maximum output current sourced by any I/O pin <sup>(3)</sup>	4 mA
Maximum current sunk by all ports	200 mA
Maximum current sourced by all ports <sup>(2)</sup>	200 mA

- **Note 1:** Stresses above those listed under "Absolute Maximum Ratings" can cause permanent damage to the device. This is a stress rating only, and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods can affect device reliability.
  - 2: Maximum allowable current is a function of device maximum power dissipation (see Table 24-2).
  - **3:** Exceptions are CLKOUT, which is able to sink/source 25 mA, and the VREF+, VREF-, SCLx, SDAx, PGECx and PGEDx pins, which are able to sink/source 12 mA.

### TABLE 24-12: DC CHARACTERISTICS: PROGRAM MEMORY

DC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V(unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +85^{\circ}C$ for Industrial					
Param No.	Symbol	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions	
		Program Flash Memory						
D130a	Eр	Cell Endurance	100	1000	—	E/W	See Note 2	
D131	Vpr	VDD for Read	VMIN	_	3.6	V	Vмın = Minimum operating voltage	
D132B	VPEW	VDD for Self-Timed Write	VMIN	_	3.6	V	Vмın = Minimum operating voltage	
D134	TRETD	Characteristic Retention	20	_	—	Year	Provided no other specifications are violated	
D135	IDDP	Supply Current during Programming	—	10	—	mA		
D136a	Trw	Row Write Time	1.32	—	1.74	ms	TRW = 11064 FRC cycles, See <b>Note 2</b>	
D136b	Trw	Row Write Time	1.28	—	1.79	ms	TRW = 11064 FRC cycles, See <b>Note 2</b>	
D137a	TPE	Page Erase Time	20.1	—	26.5	ms	TPE = 168517 FRC cycles, See Note 2	
D138a	Tww	Word Write Cycle Time	42.3	—	55.9	μs	Tww = 355 FRC cycles, See Note 2	

**Note 1:** Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

2: Other conditions: FRC = 7.37 MHz, TUN<5:0> = b'011111 (for Min), TUN<5:0> = b'100000 (for Max). This parameter depends on the FRC accuracy (see Table 24-19) and the value of the FRC Oscillator Tuning register (see Register 9-4). For complete details on calculating the Minimum and Maximum time see Section 5.3 "Programming Operations".

### TABLE 24-13: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)

Operating temperature  $-40^{\circ}C \le TA \le +85^{\circ}C$  for Industrial

Param No.	Symbol	Characteristics	Min	Тур	Max	Units	Comments
	Cefc	External Filter Capacitor Value	4.7	10	_	μF	Capacitor must be low series resistance (< 5 Ohms)

### 100-Lead Plastic Thin Quad Flatpack (PF) – 14x14x1 mm Body, 2.00 mm Footprint [TQFP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



#### Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. Chamfers at corners are optional; size may vary.

3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.

4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-110B